UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/750,885	01/05/2004	Gueorgui Bonov Chkodrov	5486-0250PUS1	2093

47973 7590 09/09/2009

WORKMAN NYDEGGER/MICROSOFT
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UT 84111

EXAMINER	
TIMBLIN, ROBERT M	
ART UNIT	PAPER NUMBER
2167	

MAIL DATE	DELIVERY MODE
09/09/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/750,885	CHKODROV ET AL.	
	Examiner	Art Unit	
	ROBERT TIMBLIN	2167	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *29 June 2009*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *8-10,13,14,22-24 and 27-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *8-10,13,14,22-24 and 27-30* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

This Office Action corresponds to application 10/750,885 filed 1/5/2004.

Response to Amendment

Applicant herein amends claims 8, 22, 29, and 30. Claims 11-12 and 25-26, have been cancelled while no new claims are added. Accordingly, claims 8-10, 13, 14, 22-24, and 27-30 are pending.

35 USC § 101

With respect to claim 29 and dependents, the claims are amended to clarify that the system is comprised of hardware (i.e. processor) and thus defines statutory subject matter. The previous rejection under section 101 is removed in light of the amendment.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 8-10, 13, 14, 22-24, and 27-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Waldorf et al. ('Waldorf' hereafter; U.S. Patent Application 2002/0038228) in view of Campbell et al. ('Campbell' hereafter; U.S. Patent 6,856,970) and further in view of Meinig (U.S. Patent 6,934,714 B2).

With respect to claim 8, Waldorf teaches A method for maintaining information about multiple instances of an activity related to a business process, comprising:

receiving process data (figures 2a-b and 0047-0048; e.g. business process data) regarding the instances (figures 2a-2b, drawing references 206-212; e.g. instances and activities of an instance) from each of a plurality of application programs (figure 1b, drawing references 166, 170, and 174, also figure 10 showing an exemplary embodiment of figure 1b);

receiving continuation data (figures 2b and 3, the fields of tables 208-212 as well as 302-304. For example, ACTIVITY_ID in the process table correlates an activity with a PROCESS_ID) regarding the instances (figures 2a-2b, drawing references 206-212; e.g. instances and activities of an instance), the continuation data (tables of figures 2-3) correlating (0055-0057), for each of the instances (0056, i.e. identifying instances of a process), process data (figures 2a-b and 0047-0048; e.g. business process data) for the instance received from at least one of the application programs (figure 1b, drawing references 166, 170, and 174) with process data (figures 2a-b and 0047-0048; e.g. business process data) for the same instance received from at least another of the application programs (figure 3 and drawing reference 240; e.g. an instance is correlated with a respective application via field 332); and

inserting process data (figures 2a-b and 0047-0048; e.g. business process data) for each of the instances (drawing reference 208) into instance database records (110 and figure 2b) based on the continuation data (tables of figures 2-3), wherein:

the instances (drawing reference 208) are acted upon in a sequence of processing steps (drawing reference 248, 0006, 0057 and figs. 1b, 10), each of the applications (figure 1b, drawing references 166, 170, and 174) provides process data (figures 2a-b and 0047-0048; e.g. business process data) corresponding to a different part of the processing sequence (0060; i.e. indicating a position for an activity within a sequential process and fig. 10; e.g. information from a portal, credit agency and shipping warehouse describe a different part of an process of ordering goods), and

process data for at least a portion of the instances are received in an order different form the processing sequence (0016; e.g. when the activity is complete, the activity performing system acknowledges the status by writing a message to the queue teaches activities are written when completed and further, 0014, 0015, 0064; describe situations (i.e. delayed processing, network traffic, and disparate remote system) that may cause data to be entered out of order); and

wherein an out-of-order record (e.g. 208) is a record that contains data reflecting the completion of a processing step for an instance (258; e.g. stop timestamp indicating completion) but does not either contain or refer to currently existing data reflecting the completion of a sequentially prior processing step (e.g. with independent applications 1030-1050, fig. 10 submitting instance data, that data may come out of order) for the instance (0014-0016, 0064 describing unordered processing);

providing access to a first instance database record for an instance not containing out-of-order data (0062, 0067; wherein Waldorf describes accessing tables and opening multiple records);

receiving correlation data indicating that the first and second records pertain to the same instance (0067 and figures 2A-2B wherein the system selects the records corresponding to the Process ID and instances are identified through the Inst_ID by creating a link).

Waldorf does not appear to expressly teach preventing access to instance database records containing out-of-order data and preventing access to a second instance database record for the instance, wherein the second instance database record contains out-of-order data, and wherein process data in the second instance database record is not correlated to process data in the first record by continuation data.

Campbell, however, teaches preventing access to instance database records containing out-of-order data and preventing access to a second instance database record for the instance, wherein the second instance database record contains out-of-order data (col. 13 line 25-32 and col. 17, line 42-47) to exclude the data that is in the process of being uploaded as well as inconsistent data from view and enabling a user to see a record in its entirety when complete and preventing access to a second instance database record for the instance (col. 17 line 47-48), wherein the second instance database record contains out-of-order data (col. 17 lines 45-64; e.g. inconsistent data), and wherein process data in the second instance database record is not correlated to process data in the first record by continuation data (col. 13 lines 32-36 wherein Campbell teaches receiving unrelated (i.e. not correlated) records which are not part of a statement).

It would have been obvious to one of ordinary skill in the data processing art (e.g. monitoring completion of a process) at the time of the present invention to combine the teachings of the cited references because the system of Campbell would have given a user of Waldorf the ability to view complete data records for workflow analysis (needed by Waldorf, 0052). Such teachings of Campbell would have further given Waldorf complete and consistent records to analyze in order for an analyst to determine the "whole picture" of a business process (need suggested in Waldorf, 0010) rather than partial data which may compromise analysis.

The combination of Waldorf and Campbell do not appear to expressly teach merging the first and second records and deleting the second record.

Meinig, however, teaches merging the first and second records (col. 15 lines 6-15) for consolidating fields in a family of records (e.g. correlated records) and deleting the second record (col. 15 lines 14 and 35-40) for deleting duplicate records.

In the same field of endeavor, (i.e. data processing and record management), it would have been obvious to one of ordinary skill in the data processing art at the time of the present invention to combine the teachings of the cited references because Meinig's teachings of merging and deleting would have given the combination of references a way to combine data into few components (needed by Waldorf, 0044) for the benefit of saving storage. Further, Meinig's teachings would have given the combination a more efficient way to separate details of a business process from enormous amounts of data (needed by Waldorf, 0044) for the benefit of an accurate analysis.

With respect to claim 9, Campbell teaches the method of claim 8, wherein said preventing access comprises preventing human users from viewing instance database records containing out-of-order data (col. 17 line 23-41, Campbell).

With respect to claim 10, Campbell teaches the method of claim 8, wherein said preventing access comprises preventing one or more display or analysis application programs from performing display of or analysis upon records containing out-of-order data (col. 13 line 12-30).

With respect to claim 13, Waldorf teaches the method of claim 8, wherein the process data is received in batch updates from the applications (0068).

With respect to claim 14, Waldorf teaches the method of claim 8, wherein:

process data from at least one of the applications is sequentially pre-sorted prior to batch update (0057 and drawing reference 248).

With respect to claim 22, Waldorf teaches A computer-readable medium having stored thereon a program for maintaining information about multiple instances of an activity related to a business process which, when executed by a processor, cause the processor to perform steps comprising:

receiving process data (figures 2a-b and 0047-0048; e.g. business process data) regarding multiple instances (0011) of an activity (drawing reference 210) from each of a plurality of application programs (figure 1B);

receiving continuation data (figures 2b and 3, the fields of tables 208-212 as well as 302-304. For example, ACTIVITY_ID in the process table correlates an activity of an instance with a PROCESS_ID) regarding the instances (figures 2a-2b, drawing references 206-212; e.g. instances and activities of an instance), the continuation data (tables of figures 2-3) correlating (0055-0057), for each of the instances (0056, i.e. identifying instances of a process), process data (figures 2a-b and 0047-0048; e.g. business process data), process data (figures 2a-b and 0047-0048; e.g. business process data) for the instance received from at least one of the application programs (figure 1b, drawing references 166, 170, and 174) with process data for the same instance received from at least another of the application programs (figure 3 and drawing reference 240; e.g. an instance is correlated with a respective application via field 332);

and inserting process data (figures 2a-b and 0047-0048; e.g. business process data) for each of the instances (drawing reference 208) into instance database records (110 and figure 2b) based on the continuation data (tables of figures 2-3), wherein: the instances are acted upon in a sequence of processing steps (drawing reference 248 and 0057), each of the applications (figure 1B) provides process data (figures 2a-b and 0047-0048; e.g. business process data) corresponding to a different part of the processing sequence (0060; i.e. indicating a position for an activity within a sequential

process and fig. 10; e.g. information from a portal, credit agency and shipping warehouse describe a different part of an process of ordering goods).

process data for at least a portion of the instances are received in an order different form the processing sequence (0016; e.g. when the activity is complete, the activity performing system acknowledges the status by writing a message to the queue teaches activities are written when completed and further, 0014, 0015, 0064; describe situations (i.e. delayed processing, network traffic, and disparate remote system) that may cause data to be entered out of order)

wherein an out-of-order record (e.g. 208) is a record that contains data reflecting the completion of a processing step for an instance (258; e.g. stop timestamp indicating completion) but does not either contain or refer to currently existing data reflecting the completion of a sequentially prior processing step (e.g. with independent applications 1030-1050, fig. 10 submitting instance data, that data may come out of order) for the instance (0014-0016, 0064 describing unordered processing).

providing access to a first instance database record for an instance not containing out-of-order data (0062, 0067; wherein Waldorf describes accessing tables and opening multiple records);

receiving correlation data indicating that the first and second records pertain to the same instance (0067 and figures 2A-2B wherein the system selects the records corresponding to the Process ID and instances are identified through the Inst_ID by creating a link)

Waldorf does not appear to expressly teach preventing access to instance database records containing out-of-order data and preventing access to a second instance database record for the instance, wherein the second instance database record contains out-of-order data, and wherein process data in the second instance database record is not correlated to process data in the first record by continuation data.

Campbell, however, teaches preventing access to instance database records containing out-of-order data and preventing access to a second instance database record for the instance, wherein the second instance database record contains out-of-order data (col. 13 line 25-32 and col. 17, line 42-47) to exclude the data that is in the process of being uploaded from view and enabling a user to see a record in its entirety when complete and preventing access to a second instance database record for the instance, wherein the second instance database record contains out-of-order data, and wherein process data in the second instance database record is not correlated to process data in the first record by continuation data (col. 13 lines 32-36 wherein Campbell teaches receiving unrelated (i.e. not correlated) records).

It would have been obvious to one of ordinary skill in the data processing art (e.g. monitoring completion of a process) at the time of the present invention to combine the teachings of the cited references because the system of Campbell would have given a user of Waldorf the ability to view complete data records for workflow analysis (needed by Waldorf, 0052). Such teachings of Campbell would have further given Waldorf complete and consistent records to analyze in order for an analyst to determine the

"whole picture" of a business process (need suggested in Waldorf, 0010) rather than partial data which may compromise analysis.

The combination of Waldorf and Campbell do not appear to expressly teach merging the first and second records and deleting the second record.

Meinig, however, teaches merging the first and second records (col. 15 lines 6-15) for consolidating fields in a family of records (e.g. correlated records) and deleting the second record (col. 15 lines 14 and 35-40) for deleting duplicate records.

In the same field of endeavor, (i.e. data processing and record management), it would have been obvious to one of ordinary skill in the data processing art at the time of the present invention to combine the teachings of the cited references because Meinig's teachings of merging and deleting would have given the combination of references a way to combine data into few components (needed by Waldorf, 0044) for the benefit of saving storage. Further, Meinig's teachings would have given the combination a more efficient way to separate details of a business process from enormous amounts of data (needed by Waldorf, 0044) for the benefit of an accurate analysis.

With respect to claim 23, Campbell teaches the computer-readable medium of claim 22, wherein said preventing access comprises preventing human users from viewing instance database records containing out-of-order data (col. 13 line 25-30).

With respect to claim 24, Campbell teaches he computer-readable medium of claim 22, wherein said preventing access comprises preventing one or more display or

analysis application programs from performing display of or analysis upon records containing out-of-order data (col. 13 line 12-30).

With respect to claim 27, Waldorf teaches the computer-readable medium of claim 22, wherein the process data is received in batch updates from the applications (0068).

With respect to claim 28, Waldorf teaches the computer-readable medium of claim 22, wherein:

process data from at least one of the applications is sequentially pre-sorted prior to batch update (0057 and drawing reference 248).

With respect to claim 29, Waldorf teaches In a system having multiple application programs providing data with respect to activities related to particular instances of a business process to a process instance database, a computing device for managing the intake of and access to said data to and from said instance database, said computing device comprising:

a first receiving unit (0070; e.g. the system receiving from an application) to receive process data (figures 2a-b and 0047-0048; e.g. business process data) regarding the instances (figures 2a-2b, drawing references 206-212; e.g. instances and activities of an instance) from each of a plurality of application programs (figure 1b, drawing references 166, 170, and 174);

a second receiving unit (110) to receive continuation data (figures 2b and 3, the fields of tables 208-212 as well as 302-304. For example, ACTIVITY_ID in the process table correlates an activity with a PROCESS_ID) regarding the instances (figures 2a-2b, drawing references 206-212; e.g. instances and activities of an instance), the continuation data (tables of figures 2-3) correlating (0055-0057), for each of the instances (0056, i.e. identifying instances of a process), process data (figures 2a-b and 0047-0048; e.g. business process data) for the instance received from at least one of the application programs (figure 1b, drawing references 166, 170, and 174) with process data (figures 2a-b and 0047-0048; e.g. business process data) for the same instance received from at least another of the application programs (figure 3 and drawing reference 240; e.g. an instance is correlated with a respective application via field 332);

an inserting unit (figure 1B; i.e. the system inserts data into instance database 110) to insert process data (figures 2a-b and 0047-0048; e.g. business process data) for each of the instances (drawing reference 208) into instance database records (110 and figure 2b) based on the continuation data (tables of figures 2-3), wherein the instance database resides, at least partially, on a computer-readable memory storage medium (0042);

a sequencing unit (0068; i.e. the system requests the activities in sequence) that tracks and manages (0013) the incoming application data (figure 1b; i.e. date from applications 166-174) for each instance so that it corresponds with a processing sequence (248) that determines an order of steps defined by a process (0057)

ordering unit (248) that identifies process data for any instance that is received in an order different from the processing sequence (0016; e.g. when the activity is complete, the activity performing system acknowledges the status by writing a message to the queue teaches activities are written when completed and further, 0014, 0015, 0064; describe situations (i.e. delayed processing, network traffic, and disparate remote system) that may cause data to be entered out of order)

wherein an out-of-order record (e.g. 208) is a record that contains data reflecting the completion of a processing step for an instance (258; e.g. stop timestamp indicating completion) but does not either contain or refer to currently existing data reflecting the completion of a sequentially prior processing step (e.g. with independent applications 1030-1050, fig. 10 submitting instance data, that data may come out of order) for the instance (0014-0016, 0064 describing unordered processing);

a correlating unit (figure 1, 110) that properly correlates database records of out-of-order processing data (figure 3 drawing reference 304 ; e.g. an In Table queue) for an instance (240) with the remaining processing data for that instance (0062, 0064; i.e. the In Table receives instance records as they are entered from the applications and therefore describes out-of-order data); and

a merging unit (130 and 0057; i.e. the accumulation of records of activities belonging to the same instance) that merges out-of-order processing data records for an instance with in-order processing data records (drawing reference 248) for an instance (0011 and 0080; i.e. the system 100 collects for the individual instances) where

the in-order and out-of-order data records are correlated by said correlating unit (110; i.e. the instance database stores the instance data such as found in figure 2B).

Waldorf does not appear to teach a limiting unit that prevents access to instance database records containing out-of-order data.

Campbell, however, teaches a limiting unit (col. 13 line 26; i.e. a gatekeeper mechanism) that prevents access to instance database records containing out-of-order data (col. 13 line 25-32 and col. 17, line 42-47) to exclude the data that is in the process of being uploaded from view and enabling a user to see a record in its entirety when complete.

It would have been obvious to one of ordinary skill in the data processing art (e.g. monitoring completion of a process) at the time of the present invention to combine the teachings of the cited references because the system of Campbell would have given a user of Waldorf the ability to view complete data records for workflow analysis (needed by Waldorf, 0052). Such teachings of Campbell would have further given Waldorf complete and consistent records to analyze in order for an analyst to determine the "whole picture" of a business process (need suggested in Waldorf, 0010) rather than partial data which may compromise analysis,

wherein at least one of said inserting unit, sequencing unit, ordering unit, limiting unit, correlating unit and merging unit includes a processor (0041; e.g. microprocessor implements the system).

Waldorf and Campbell do not appear to expressly teach deleting said out-of-order records after merging.

Meinig, however, teaches deleting said out-of-order records (col. 15 lines 14 and 35-40) after merging (col. 15 lines 6-15) in a consolidation process.

In the same field of endeavor, (i.e. data processing and record management), it would have been obvious to one of ordinary skill in the data processing art at the time of the present invention to combine the teachings of the cited references because Meinig's teachings of merging and deleting would have given the combination of references a way to combine data into few components (needed by Waldorf, 0044) for the benefit of saving storage. Further, Meinig's teachings would have given the combination a more efficient way to separate details of a business process from enormous amounts of data (needed by Waldorf, 0044) for the benefit of an accurate analysis.

With respect to claim 30, Waldorf teaches A method for maintaining information about multiple instances, of an activity related to a business process, comprising:

receiving process data (figures 2a-b and 0047-0048; e.g. business process data) regarding the instances (figures 2a-2b, drawing references 206-212; e.g. instances and activities of an instance) from each of a plurality of application programs (figure 1b, drawing references 166, 170, and 174);

receiving continuation data (figures 2b and 3, the fields of tables 208-212 as well as 302-304. For example, ACTIVITY_ID in the process table correlates an activity with a PROCESS_ID) regarding the instances (figures 2a-2b, drawing references 206-212; e.g. instances and activities of an instance), the continuation data (tables of figures 2-3) correlating (0055-0057), for each of the instances (0056, i.e. identifying instances of a

process), process data (figures 2a-b and 0047-0048; e.g. business process data) for the instance received from at least one of the application programs (figure 1b, drawing references 166, 170, and 174) with process data (figures 2a-b and 0047-0048; e.g. business process data) for the same instance received from at least another of the application programs (figure 3 and drawing reference 240; e.g. an instance is correlated with a respective application via field 332); and

inserting process data (figures 2a-b and 0047-0048; e.g. business process data) for each of the instances (drawing reference 208) into instance database records (110 and figure 2b) based on the continuation data (tables of figures 2-3), wherein each instance database record comprises a primary key (240), a timestamp field for when the data was received (256), a field noting the geographical origin of the data (254 and 0058; i.e. the Who field may indicate the application which made the entry. Figure 10 shows the multiple applications (1030-1050) and their locations that may be included in that field), a field noting the size of the data (242), a field noting the time the data was collected (0055), and field noting the time the data was transmitted (0068; e.g. then the activity was requested), and further wherein:

the instances (drawing reference 208) are acted upon in a sequence of processing steps (drawing reference 248 and 0057), each of the applications (figure 1b) provides process data (figures 2a-b and 0047-0048; e.g. business process data) corresponding to a different part of the processing sequence (0060; i.e. indicating a position for an activity within a sequential process)

process data for at least a portion of the instances are received in an order different from the processing sequence (0016; e.g. when the activity is complete, the activity performing system acknowledges the status by writing a message to the queue teaches activities are written when completed and further, 0014, 0015, 0064; describe situations (i.e. delayed processing, network traffic, and disparate remote system) that may cause data to be entered out of order); and

wherein an out-of-order record (e.g. 208) is a record that contains data reflecting the completion of a processing step for an instance (258; e.g. stop timestamp indicating completion) but does not either contain or refer to currently existing data reflecting the completion of a sequentially prior processing step (e.g. with independent applications 1030-1050, fig. 10 submitting instance data, that data may come out of order) for the instance (0014-0016, 0064 describing unordered processing) and further wherein said out-of-order data records have not been correlated (e.g. figure 3) with the appropriate existing data records, said out-of-order records being similarly comprised to said instance database records (fig. 2b).

Waldorf does not appear to expressly teach preventing access to instance database records containing out-of-order data.

Campbell, however, teaches preventing access to instance database records containing out-of-order data (col. 13 line 25-32 and col. 17, line 42-47) to exclude the data that is in the process of being uploaded from view and enabling a user to see a record in its entirety when complete.

It would have been obvious to one of ordinary skill in the data processing art (e.g. monitoring completion of a process) at the time of the present invention to combine the teachings of the cited references because the system of Campbell would have given a user of Waldorf the ability to view complete data records for workflow analysis (needed by Waldorf, 0052). Such teachings of Campbell would have further given Waldorf complete and consistent records to analyze in order for an analyst to determine the "whole picture" of a business process (need suggested in Waldorf, 0010) rather than partial data which may compromise analysis.

Waldorf and Campbell do not appear to expressly teach correlating said out-of-order records with said appropriate existing data records, where correlating includes merging the data of said out-of-order records with said existing records and then deleting said out-of-order records.

Meinig, however, teaches correlating said out-of-order records (col. 1 lines 57-60; e.g. identifying a family of records) with said appropriate existing data records (col. 15 lines 6-16 wherein certain fields of an existing record are consolidated), where correlating includes merging (col. 15 lines 6-15) the data of said out-of-order records with said existing records and then deleting said out-of-order records (col. 15 lines 14 and 35-40) for consolidating records and deleting duplicates.

In the same field of endeavor, (i.e. data processing and record management), it would have been obvious to one of ordinary skill in the data processing art at the time of the present invention to combine the teachings of the cited references because Meinig's teachings of merging and deleting would have given the combination of references a

way to combine data into few components (needed by Waldorf, 0044) for the benefit of saving storage. Further, Meinig's teachings would have given the combination a more efficient way to separate details of a business process from enormous amounts of data (needed by Waldorf, 0044) for the benefit of an accurate analysis.

Response to Arguments

Applicant's arguments with respect to claims 8 and similar claims 22, 29 and 30 have been considered but are moot in view of the new ground(s) of rejection.

With respect to the arguments pertaining to Waldorf and Campbell failing to teach merging and then deleting (e.g. page 10 of remarks), Examiner agrees. That is, Waldorf teaches data relation rather than data merging. Campbell is maintained to teach the aspect of preventing access to an instance record containing out of order data.

However, with respect to the clarified merging and then deleting aspect, Examiner submits that the cited Meinig reference teaches said aspect as applied in the rejection above. Therefore, the argument is considered moot in view of the new ground of rejection.

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Contact Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Robert M. Timblin whose telephone number is 571-272-5627. The examiner can normally be reached on M-TH 8:00-5:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John R. Cottingham can be reached on 571-272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/ROBERT TIMBLIN/
Examiner, Art Unit 2167

/John R. Cottingham/
Supervisory Patent Examiner, Art Unit 2167